1

Blockchain 101

It is very likely that if you are reading this book, you already have heard about blockchain and have some fundamental appreciation of its enormous potential. If not, then let me tell you that this is a technology that has promised to positively alter the existing paradigms of nearly all industries, including, but not limited to, the IT, finance, government, media, medical, and law sectors, by making them more efficient, trustworthy, and transparent.

This chapter introduces blockchain technology, its technical foundations, the theory behind it, and various technologies that have contributed to building what is known today as blockchain. The theoretical foundations of distributed systems are described first. Next, the precursors of Bitcoin are presented. Finally, blockchain technology is introduced. This approach is a logical way of understanding blockchain technology, as the roots of blockchain are in distributed systems and cryptography. We will be covering a lot of ground quickly here, but don't worry—we will go over a great deal of this material in much greater detail as you move throughout the book.

In this chapter, we'll focus on:

- The growth of blockchain technology
- Distributed systems
- The history of blockchain
- Introducing blockchain
- Types of blockchain

The growth of blockchain technology

With the invention of Bitcoin in 2008, the world was introduced to a new concept that revolutionized the whole of society. It was something that promised to have an impact upon every industry. This new concept was blockchain, the underlying technology that underpins Bitcoin.

Some describe blockchain as a revolution, whereas another school of thought believes that it is going to be more evolutionary, and it will take many years before any practical benefits of blockchain reach fruition. This thinking is correct to some extent, but, in my opinion, the revolution has already begun.

It is a technology that has an impact on current technologies too and possesses the ability to change them at a fundamental level.

Progress towards maturity

Around 2013, some ideas emerged regarding using blockchain technology for other applications instead of only cryptocurrencies. Then, in 2014, some research and experimentation began, which led to proofs of concept, further research, and full-scale trial projects between 2015 and 2017. In 2015, Ethereum was launched as the first smart contract programmable blockchain, which unlocked many possibilities. Interest in enterprise-grade blockchains originated around the same time. Around that time, we also saw several projects, such as Everledger, Quorum, and Corda.

In addition, the development of novel monolithic and multichain architectures rapidly evolved after 2018. Some key examples of novel protocols include Solana, Avalanche, and Polkadot, which we will discuss later in this book.

Currently, as of the second quarter of 2023, **decentralized finance** (**DeFi**), n**on-fungible tokens** (**NFTs**), and tokenization in general are very popular applications of blockchain. They are already in mainstream usage by millions of users around the world who are engaging in DeFi services. It is expected that within three years or so, DeFi will stabilize into a mature mainstream technology. With all the activity and adoption in DeFi and NFT trading, we can say that to some extent blockchain is already part of our daily lives. Of course, further maturity is required, especially from a regulation and security perspective, but millions of users are already using blockchain regularly, either to make payments, trade NFTs, get loans, or play games.

> See the following article for details on which platforms these millions are making use of: `https://beincrypto.com/ethereum-defi-users-reach-new-highs-over-4m-growing-roughly-8x-in-a-year/`

This trend is only expected to grow and, during 2022 and onwards, more and more research and development will be carried out. There will be more focus on the regulation and standardization of blockchain technology. Numerous projects are already production ready, and more adoption is expected in the coming years.

> Progress in blockchain technology almost feels like the internet dot-com boom of the late 1990s.

Research in the scalability of blockchains, where blockchains can handle many transactions like traditional financial networks, has led to the development of **layer 2 architectures** and **multi-chain architectures**. Advancement in making zero-knowledge proofs practical helped tremendously in making layer 2 solutions a reality. Layer 2 solutions are under heavy research and development now, and many mechanisms have been introduced, such as Plasma, rollups, sidechains, Lightning Network, and many others.

As of 2022, blockchain is already something that many people use every day. With DeFi, NFTs, cryptocurrencies, and Metaverses, blockchain has attracted millions of users. Most of the attraction is because of financial incentives gained by the trading and investment of digital assets such as NFTs and other DeFi services; however, researchers and academics are also interested in studying and developing the theory of these applications and relevant protocols.

Undoubtedly, further maturation and the adoption of blockchain technology is expected in the coming years.

Rising interest

Interest in blockchain technology has risen quite significantly over the last few years. Once dismissed by some simply as "geek money" from a cryptocurrency point of view or as something that was just not considered worth pursuing, blockchain is now being researched by the largest companies and organizations around the world. Millions of dollars are being spent to adopt and experiment with this technology.

Also, the interest in blockchain within academia is astounding, and many educational establishments—including prestigious universities around the world—are conducting research and development on blockchain technology. There are not only educational courses being offered by many institutions, but academics are also conducting high-quality research and producing many insightful research papers on the topic. There are also several research groups and conferences worldwide that specifically focus on blockchain research. This research community is beneficial for the growth of the entire blockchain ecosystem. A simple online search of "blockchain research groups" would reveal hundreds, if not thousands, of these research groups.

> There are also various consortiums, such as **Enterprise Ethereum Alliance** (**EEA**) at `https://entethalliance.org` and Hyperledger at `https://www.hyperledger.org`, that have been established for the research, development, and standardization of blockchain technology. Moreover, the **Institute of Electrical and Electronics Engineers** (**IEEE**) and the **International Standards Organization** (**ISO**) have also started their attempts to standardize various aspects of blockchain technology.

Many start-ups are providing blockchain-based solutions already. A simple trend search on Google reveals the immense scale of interest in blockchain technology over the last few years. Hot topics include **decentralized autonomous organizations** (**DAOs**), fully autonomous and transparent member-governed entities developed using smart contracts with no central authority. Meanwhile, NFTs, through which digital art is routinely bought and sold for millions of dollars, and Metaverses have been making the news.

> A Metaverse is a computer-simulated three-dimensional environment. A convergence of our digital and real-world life, they provide a virtual world for the users to perform activities such as social interactions, engaging in business, and shopping. This virtual world is usually accessible via specialized hardware called VR headsets, enhancing the virtual world experience.
>
> This is not a new concept; in the Web 2.0 days (the usual internet that we know and use daily), Second Life, World of Warcraft, and quite a few other similar platforms rose to prominence, as entities centrally owned by shareholders. In the Web 3.0 era (post blockchain), Decentraland and the Sandbox, among many others, are becoming popular, based on decentralized foundations and community governance.

In this book, we are going to learn what exactly blockchain technology is and how it can reshape businesses, multiple industries, and indeed everyday life by bringing about a plenitude of benefits such as decentralized trust, efficiency, cost savings, transparency, and security. We will also explore what decentralization is, smart contracts, and how solutions can be developed and implemented using blockchain platforms such as Ethereum.

While there are many benefits of blockchain technology, there are some challenges too that can cause hurdles in adoption and are being actively researched, such as scalability, privacy, and security. We'll also take a critical look at blockchain and discuss its limitations and challenges.

We shall begin our exploration of blockchain by looking at distributed systems in the following section. This is a foundational paradigm on which blockchain is based, and we must have a good grasp of what distributed systems are before we can meaningfully discuss blockchain in detail.

Distributed systems

Understanding distributed systems is essential to our understanding of blockchain, as blockchain is a distributed system at its core. It is a distributed ledger that can be centralized or decentralized. A blockchain is originally intended to be and is usually used as a decentralized platform. It can be thought of as a system that has properties of both decentralized and distributed paradigms. It is a decentralized-distributed system.

A **distributed system** is a computing paradigm whereby two or more nodes work with each other in a coordinated fashion to achieve a common outcome. It is modeled in such a way that end users see it as a single logical platform. For example, Google's search engine is based on a large distributed system; however, to a user, it looks like a single, coherent platform. It is composed of processes (nodes) and channels (communication channels) where nodes communicate by passing messages. A blockchain is a message-passing distributed system.

A **node** is an individual player (a computer) in a distributed system. All nodes can send and receive messages to and from each other. Nodes can be honest, faulty, or malicious, and they have memory and a processor. A node that exhibits arbitrary behavior is known as a *Byzantine node* after the **Byzantine Generals** problem.

The Byzantine Generals problem

In 1982, a thought experiment was proposed by Lamport et al. in their research paper, *The Byzantine Generals Problem*, which is available here: `https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/`

In this problem, a group of army generals who lead different parts of the Byzantine army is planning to attack or retreat from a city. The only way of communicating with them is via a messenger. They need to agree to strike at the same time to win. The issue is that one or more generals might be traitors who could send a misleading message. Moreover, the messenger could be captured by the city, resulting in no message delivery. Therefore, there is a need for a viable mechanism that allows agreement among the generals, even in the presence of the treacherous ones, and message loss, so that the attack can still take place at the same time. As an analogy for distributed systems, the generals can be considered as honest nodes, the traitors as Byzantine nodes (that is, nodes with arbitrary behavior), the messenger can be thought of as a channel of communication with the generals, and a captured messenger as a delayed or lost message. Several solutions were presented to this problem in the paper by Lamport et al. in 1982.

This type of inconsistent behavior of Byzantine nodes can be intentionally malicious, which is detrimental to the operation of the network. Any unexpected behavior by a node on the network, whether malicious or not, can be categorized as Byzantine.

A small-scale example of a distributed system is shown in the following diagram. This distributed system has six nodes, of which one (*N4*) is a Byzantine node, leading to possible data inconsistency. *L2* is a link that is broken or slow, and this can lead to a partition in the network:

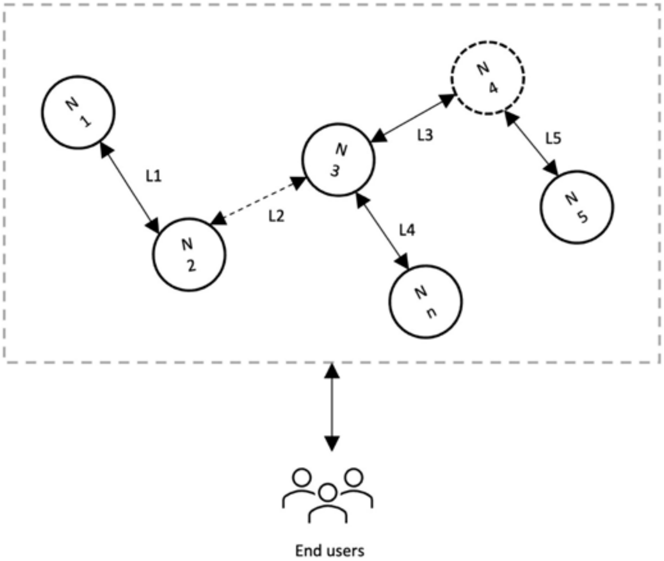

Figure 1.1: Design of a distributed system: N4 is a Byzantine node and L2 is broken or a slow network link

Two key challenges of a distributed system design are the coordination between nodes and fault tolerance. Even if some (a certain threshold dictated by the consensus protocol) of the nodes become faulty or network links break, the distributed system should be able to tolerate this and continue to work to achieve the desired result. This problem has been an active area of distributed system design research for many years, and several algorithms and mechanisms have been proposed to overcome these issues.

Distributed systems are challenging to design. It has been proven that a distributed system cannot have all three of the much-desired properties of consistency, availability, and partition tolerance simultaneously. This principle is known as the **CAP theorem**.

CAP theorem

The **CAP theorem**, also known as Brewer's theorem, was introduced by Eric Brewer in 1998 as a conjecture. In 2002, it was proven as a theorem by Seth Gilbert and Nancy Lynch. The theorem states that any distributed system cannot have consistency, availability, and partition tolerance simultaneously:

- **Consistency** is a property that ensures that all nodes in a distributed system have a single, current, and identical copy of the data. Consistency is achieved using consensus algorithms to ensure that all nodes have the same copy of the data. This is also called **state machine replication** (**SMR**). The blockchain is a means of achieving state machine replication.
- **Availability** means that the nodes in the system are up, accessible, and are accepting incoming requests and responding with data without any failures as and when required. In other words, data is available at each node and the nodes are responding to requests.
- **Partition tolerance** ensures that if a group of nodes is unable to communicate with other nodes due to network failures, the distributed system continues to operate correctly. This can occur due to network and node failures.

A Venn diagram is commonly used to visualize the CAP theorem, as shown below:

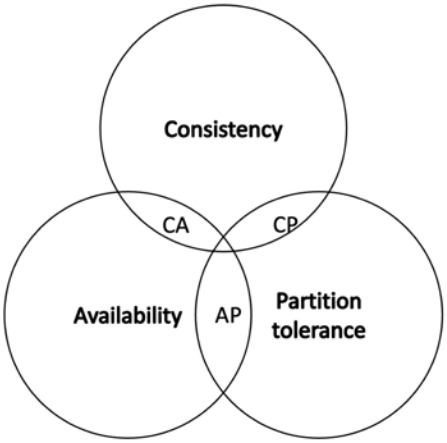

Figure 1.2: CAP theorem

The preceding diagram depicts that only two properties at a time can be attained; either *AP*, *CA*, or *CP*.

In summary:

- If we opt for *CP* (consistency and partition tolerance), we sacrifice availability.
- If we opt for *AP* (availability and partition tolerance), we sacrifice consistency.
- If we opt for *AC* (availability and consistency), we sacrifice partition tolerance.

Note that *AC* does not really exist. The CAP theorem in practice means that in the case of a network partition, a distributed system is either available or consistent. As network partitions cannot be ignored, the choice is between consistency or availability when a network partition occurs.

We can explain this concept with the following example.

Let's imagine that there is a distributed system with two nodes. Now, let's apply the three theorem properties to this smallest of possible distributed systems with only two nodes:

- **Consistency** is achieved if both nodes have the same shared state; that is, they have the same up-to-date copy of the data.
- **Availability** is achieved if both nodes are up and running and responding with the latest copy of data.
- **Partition tolerance** is achieved if, despite communication failure or delay between nodes, the network (distributed system) continues to operate.

Now think of a scenario where a partition occurs and nodes can no longer communicate with each other. If newly updated data comes in now, it can only be updated on one node. If the node accepts the update, then only one node in the network is updated, and consistency is lost. If the node rejects the update, that will result in a loss of availability. This means that either availability or consistency is unachievable due to the network partition. This is strange because somehow, blockchain manages to achieve all these properties, violating the theorem (especially in its most successful implementation, Bitcoin)—or does it?

In blockchains, consistency is sacrificed in favor of availability and partition tolerance. In this scenario, consistency (C) in the blockchain is not achieved simultaneously with partition tolerance (P) and availability (A), but it is achieved over time. This is called eventual consistency, where consistency is achieved due to validation from multiple nodes over time. There can be a temporary disagreement between nodes on the final state, but it is eventually agreed upon. For example, in Bitcoin, multiple transaction confirmations are required to achieve a good confidence level that transactions may not be rolled back in the future. Eventually, a consistent view of transaction history is available in all nodes. Multiple confirmations of a transaction over time provide eventual consistency in Bitcoin. For this purpose, the process of mining was introduced in Bitcoin. Mining is a process that facilitates the achievement of consensus by using the **Proof of Work** (**PoW**) algorithm. At a higher level, mining can be defined as a process that's used to add more blocks to the blockchain. We will cover more on this later in *Chapter 6, Bitcoin Architecture*.

PACELC theorem

An extension of the CAP theorem called PACELC was first proposed by Daniel J. Abadi from Yale University. It states that, in addition to the three properties proposed by CAP, there are also tradeoffs between latency and consistency. It states that tradeoffs between consistency and latency always exist in replicated systems, whereas CAP is only applicable when there are network partitions. In other words, it means that even if no network partitions occur, under normal operation the tradeoff between consistency and latency exists. For example, some databases may choose to give up consistency for lower latency, and some databases could pay the availability and latency costs to achieve consistency. This is true for replicated systems and presents a more inclusive picture of consistency tradeoffs in distributed systems.

> PACELC was formally proved in a paper available here: `https://dl.acm.org/doi/10.1145/3197406.3197420`

With a better understanding of distributed systems, let's now talk about blockchain itself. First, we'll begin with a brief rundown of the history of blockchain and Bitcoin.

The history of blockchain

Blockchain was introduced with the invention of Bitcoin in 2008. Its practical implementation then occurred in 2009. Bitcoin will be explored in great depth in *Chapter 6*, *Bitcoin Architecture*. However, it is essential to refer to Bitcoin here because without it, the history of blockchain is not complete.

Now we will look at the early history of computing and computer networks and will discuss how these technologies evolved and contributed to the development of Bitcoin in 2008:

- 1976 – Diffie–Hellman work on securely exchanging cryptographic keys.
- 1978 – Invention of public key cryptography.
- 1979 – Invention of Merkle trees (hashes in a tree structure) by Ralph C. Merkle.
- 1980s – Development of TCP/IP.
- 1980 – Protocols for public key cryptosystems, Ralph C. Merkle.
- 1982 – Blind signatures proposed by David Chaum.
- 1982 – The Byzantine Generals problem.
- 1985 – Work on elliptic curve cryptography by Neal Koblitz and Victor Miller.
- 1991 – Haber and Stornetta work on tamper-proofing document timestamps. This can be considered the earliest idea of a chain of blocks or hash chains.
- 1992 – Cynthia Dwork and Moni Naor publish *Pricing via Processing or Combatting Junk Mail*. This is considered the first use of **PoW**.
- 1993 – Haber, Bayer, and Stornetta upgraded the tamper-proofing of document timestamps system with Merkle trees.

- 1995 – David Chaum's Digicash system (an anonymous electronic cash system) started to be used in some banks.
- 1998 – Bit Gold, a mechanism for decentralized digital currency, invented by Nick Szabo. It used hash chaining and Byzantine Quorums.
- 1999 – Emergence of a file-sharing application mainly used for music sharing, Napster, which is a P2P network, but was centralized with the use of indexing servers.
- 1999 – Development of a secure timestamping service for the Belgian project TIMESEC.
- 2000 – Gnutella file-sharing network, which introduced decentralization.
- 2001 – Emergence of BitTorrent and **Distributed Hash Tables** (**DHTs**).
- 2002 – Hashcash by Adam Back.
- 2004 – Development of B-Money by Wei Dei using Hashcash.
- 2004 – Hal Finney, the invention of the reusable PoW system.
- 2005 – Prevention of Sybil attacks by using computation puzzles, due to James Aspnes et al.
- 2009 – Bitcoin (first blockchain).

These technologies contributed in some way to the development of Bitcoin, even if not directly; the work is relevant to the problem that Bitcoin solved.

Bitcoin

All previous attempts to create anonymous and decentralized digital currency were successful to some extent, but they could not solve the problem of preventing double spending in a completely trustless or permissionless environment. This problem was finally addressed by the Bitcoin blockchain, which introduced the Bitcoin cryptocurrency.

Bitcoin also solves the **SMR problem**, introduced in 1978 by Leslie Lamport and formalized in 1980 by Fred Schneider. SMR is a scheme that's used to implement a fault-tolerant service by replicating data (state) between nodes in a distributed system. Bitcoin solves the problem by allowing the replication of blocks at all correct nodes and ensuring consistency via its PoW mechanism. Here, the agreement is reached between nodes (or replicas) repeatedly to append new blocks to the blockchain.

Electronic cash

The concept of **electronic cash** (**e-cash**), or digital currency, is not new. Since the 1980s, e-cash protocols have existed that are based on a model proposed by David Chaum.

Just as understanding the concepts of distributed systems is necessary to comprehend blockchain technology, the idea of e-cash is also essential to appreciate the first, and astonishingly successful, application of blockchain, Bitcoin, and more broadly, cryptocurrencies in general. To create an effective e-cash system, two fundamental requirements need to be met: **accountability** and **anonymity**.

Accountability is required to ensure that cash is spendable only once (addressing the double-spending problem) and that it can only be spent by its rightful owner. The double-spending problem arises when the same money is spent twice. As it is quite easy to make copies of digital data, this becomes a big issue in digital currencies as you can make many copies of the same digital cash. Spending the same cash twice is known as the double-spending problem.

Anonymity is required to protect users' privacy. With physical cash, it is almost impossible to trace back spending to the individual who actually paid the money, which provides adequate privacy should the consumer choose to hide their identity. In the digital world, however, providing such a level of privacy is difficult due to inherent personalization, tracing, and logging mechanisms in digital payment systems such as credit card payments. This is a required feature for ensuring the security and safety of the financial network, but it is also often seen as a breach of privacy.

This is because end users do not have any control over who their data might be shared with, even without their consent. Nevertheless, this is a solvable problem, and cryptography is used to address such issues. Especially in blockchain networks, the privacy and anonymity of the participants on the blockchain are sought-after features. David Chaum solved both problems during his work in the 1980s by using two cryptographic operations, namely, **blind signatures** and **secret sharing**. These terms and related concepts will be discussed in detail in *Chapter 4, Asymmetric Cryptography*. For the moment, it is sufficient to say that *blind signatures* allow the signing of a document without actually seeing it, and a *secret sharing* scheme enables the detection of double-spending.

In 2009, the first practical implementation of an e-cash system named Bitcoin appeared. The term cryptocurrency emerged later. For the very first time, it solved the problem of distributed consensus in a trustless network. It used **public key cryptography** with a PoW mechanism to provide a secure and decentralized method of minting digital currency. The key innovation is the idea of an ordered list of blocks composed of transactions that is cryptographically secured by the PoW mechanism to prevent double-spending in a trustless environment. This concept will be explained in greater detail in *Chapter 6, Bitcoin Architecture*.

Looking at all the technologies mentioned previously and their relevant history, it is easy to see how concepts from e-cash schemes and distributed systems were combined to create Bitcoin and what now is known as blockchain. This concept can also be visualized with the help of the following diagram:

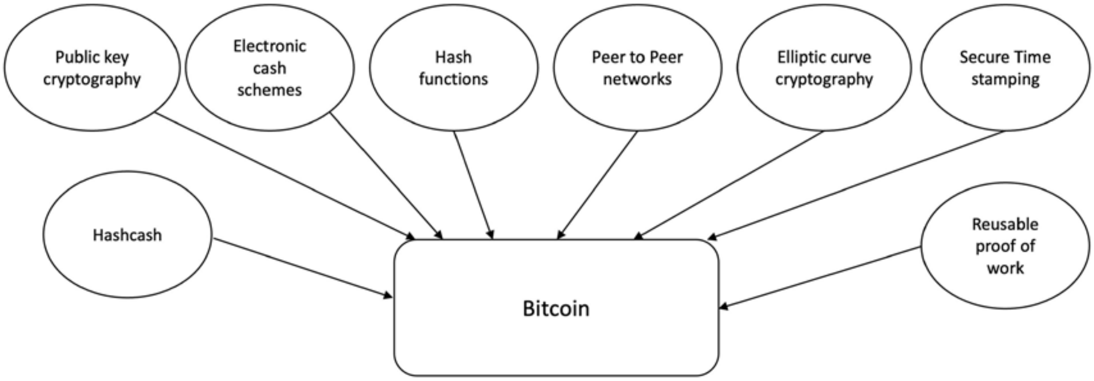

Figure 1.3: The various ideas that supported the invention of Bitcoin and blockchain

With the emergence of e-cash covered, along with the ideas that led to the formation of Bitcoin and blockchain, we can now begin to discuss blockchain itself.

Introducing blockchain

In 2008, a groundbreaking paper, entitled *Bitcoin: A Peer-to-Peer Electronic Cash System*, was written on the topic of peer-to-peer e-cash under the pseudonym of *Satoshi Nakamoto*.

> No one knows the actual identity of Satoshi Nakamoto. After introducing Bitcoin in 2009, Nakamoto remained active in the Bitcoin developer community until 2011, before handing over Bitcoin development to its core developers and simply disappearing.

The paper introduced the term **chain of blocks**, later to evolve into "blockchain", where a chronologically ordered sequence of blocks containing transactions is produced by the protocol. The paper is available at `https://bitcoin.org/bitcoin.pdf`.

There are some different ways that blockchain may be defined; the following are two of the most widely accepted definitions:

- **Layman's definition**: Blockchain is an ever-growing, secure, shared recordkeeping system in which each user of the data holds a copy of the records, which can only be updated if a majority of parties involved in a transaction agree to update.
- **Technical definition**: Blockchain is a peer-to-peer, distributed ledger that is cryptographically secure, append-only, immutable (extremely hard to change), and updateable only via consensus among peers.

Now, let's examine things in some more detail. We will look at the keywords from the technical definition one by one:

- **Peer-to-peer**: The first keyword in the technical definition is **peer-to-peer**, or **P2P**. This means that there is no central controller in the network, and all participants (nodes) talk to each other directly. This property allows transactions to be conducted directly among the peers without third-party involvement, such as by a bank.
- **Distributed ledger**: Dissecting the technical definition further reveals that blockchain is a "distributed ledger," which means that a ledger is spread across the network among all peers in the network, and each peer holds a copy of the complete ledger.
- **Cryptographically secure**: Next, we see that this ledger is "cryptographically secure," which means that cryptography has been used to provide security services that make this ledger secure against tampering and misuse. These services include non-repudiation, data integrity, and data origin authentication.
- **Append-only**: Another property that we encounter is that blockchain is "append-only," which means that data can only be added to the blockchain in *time-sequential order*. This property implies that once data is added to the blockchain, it is almost impossible to change that data and it can be considered practically immutable. In other words, blocks added to the blockchain cannot be changed, which allows the blockchain to become an immutable and tamper-proof ledger of transactions.

> A blockchain can be changed in rare scenarios where collusion against the blockchain network by bad actors succeeds in gaining more than 51% of the authority. There may also be some legitimate reasons to change data in the blockchain once it has been added, such as the “right to be forgotten” or “right to erasure” (also defined in the GDPR ruling: `https://gdpr-info.eu/art-17-gdpr/`). The right to be forgotten is the right that mandates personal data about a person to be removed from internet records, organizational records, and other associated services. However, those are individual cases that need to be handled separately and that require an elegant technical solution.

- **Updatable via consensus**: The most critical attribute of a blockchain is that it is updateable only via consensus. This is what gives it the power of decentralization. In this scenario, no central authority is in control of updating the ledger. Instead, any update made to the blockchain is validated against strict criteria defined by the blockchain protocol and added to the blockchain only after consensus has been reached among a majority of participating peers/nodes on the network. To achieve consensus, there are various consensus algorithms that ensure all parties agree on the final state of the data on the blockchain network and resolutely agree upon it to be true.

Having detailed the primary features of blockchain, we are now able to begin to look at its actual architecture.

Blockchain architecture

We’ll begin by looking at how blockchain acts as a layer within a distributed peer-to-peer network.

Blockchain by layers

Blockchain can be thought of as a layer of a distributed peer-to-peer network running on top of the internet, as can be seen in the following diagram. It is analogous to SMTP, HTTP, or FTP running on top of TCP/IP:

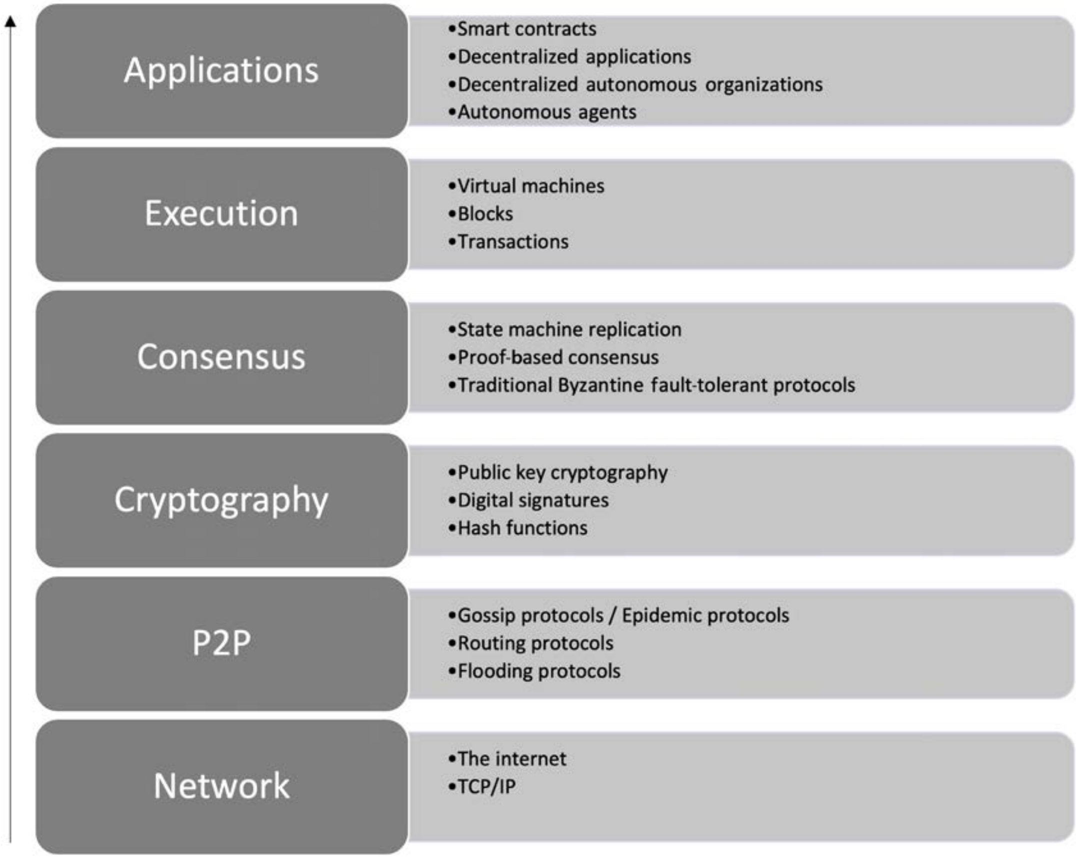

Figure 1.4: The layered architectural view of a generic blockchain

Now we'll discuss all these elements one by one:

- The lowest layer is the **Network** layer, which is usually the internet and provides a base communication layer for any blockchain.
- A **P2P** (peer-to-peer) network runs on top of the **Network** layer, which consists of information propagation protocols such as gossip or flooding protocols.
- After this comes the **Cryptography** layer, which contains crucial cryptographic protocols that ensure the security of the blockchain. These cryptographic protocols play a vital role in the integrity of blockchain processes, secure information dissemination, and blockchain consensus mechanisms. This layer consists of public key cryptography and relevant components such as digital signatures and cryptographic hash functions. Sometimes, this layer is abstracted away, but it has been included in the diagram because it plays a fundamental role in blockchain operations.

- Next comes the **Consensus** layer, which is concerned with the usage of various consensus mechanisms to ensure agreement among different participants of the blockchain. This is another crucial part of the blockchain architecture, which consists of various techniques such as SMR, proof-based consensus mechanisms, or traditional (from traditional distributed systems research) Byzantine fault-tolerant consensus protocols.
- We then have the **Execution** layer, which can consist of virtual machines, blocks, transactions, and smart contracts. This layer, as the name suggests, provides execution services on the blockchain, and performs operations such as value transfer, smart contract execution, and block generation. Virtual machines such as **Ethereum Virtual Machine** (**EVM**), **Ethereum WebAssembly** (**ewasm**), and Zinc VM provide an execution environment for smart contracts to execute.
- Finally, we have the **Applications** layer, which is composed of smart contracts, decentralized applications, DAOs, and autonomous agents. This layer can effectively contain all sorts of various user-level agents and programs that operate on the blockchain. Users interact with the blockchain via decentralized applications.

All these concepts will be discussed in detail later in this book in various chapters. Next, we'll look at blockchain from more of a business-oriented perspective.

Blockchain in business

The current traditional business model is centralized. For example, for cash transfers, banks act as a central trusted third party. In financial trading, a central clearing house acts as a trusted third party between two or more trading parties. From a business standpoint, a blockchain can be defined as a platform where peers can exchange value using transactions without the need for a centrally trusted arbitrator (a trusted third party). This concept is compelling, and, once you absorb it, you will realize the enormous potential of blockchain technology. This disintermediation allows blockchain to be a decentralized mechanism where no single authority controls the network. Immediately, we can see a significant benefit of decentralization here, because if no banks or central clearing houses are required, then it naturally leads to cost savings, faster transaction speeds, transparency, and more trust. Moreover, in the payment business, blockchain can be used to facilitate cross-border and local payments in a decentralized and secure manner.

We've now looked at what blockchain is at a fundamental level. Next, we'll go a little deeper and look at some of the elements that comprise a blockchain.

Generic elements of a blockchain

Now, let's walk through the generic elements of a blockchain. You can use this as a handy reference section if you ever need a reminder about the different parts of a blockchain. More precise elements will be discussed in the context of their respective blockchains in later chapters, for example, the Ethereum blockchain. The structure of a generic blockchain can be visualized with the help of the following diagram:

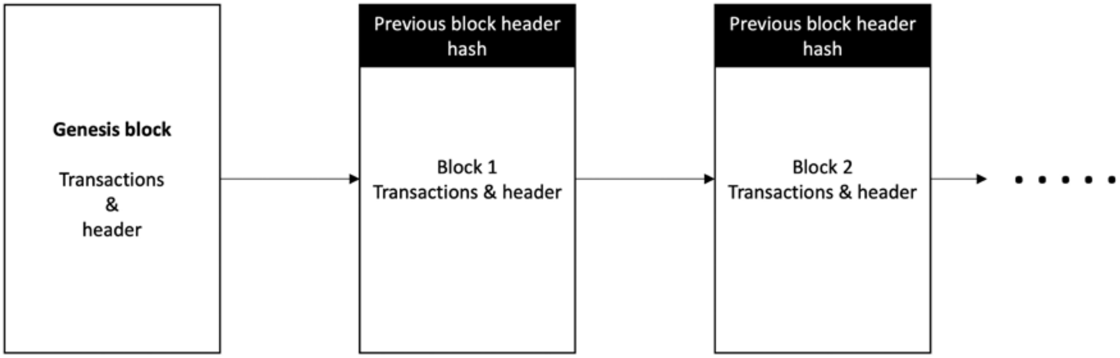

Figure 1.5: Generic structure of a blockchain

Elements of a generic blockchain are described here one by one. These are the elements that you will come across in relation to blockchain:

- **Address**: Addresses are unique identifiers used in a blockchain transaction to denote senders and recipients. An address is usually a public key or derived from a public key.
- **Transaction**: A transaction is the fundamental unit of a blockchain. A transaction represents a transfer of value from one address to another.
- **Block**: A block is composed of multiple transactions and other elements, such as the previous block hash (hash pointer), timestamp, and nonce. A block is composed of a block header and a selection of transactions bundled together and organized logically. A block contains several elements, which we introduce as follows:
    - A reference to a previous block is also included in the block unless it is a genesis block. This reference is the hash of the header of the previous block. A **genesis block** is the first block in the blockchain that is hardcoded at the time the blockchain was first started. The structure of a block is also dependent on the type and design of a blockchain.
    - A **nonce** is a number that is generated and used only once. A nonce is used extensively in many cryptographic operations to provide replay protection, authentication, and encryption. In blockchain, it's used in PoW consensus algorithms and for transaction replay protection. A block also includes the nonce value.
    - A **timestamp** is the creation time of the block.
    - **Merkle root** is a hash of all the nodes of a Merkle tree. In a blockchain block, it is the combined hash of the transactions in the block. Merkle trees are widely used to validate large data structures securely and efficiently. In the blockchain world, Merkle trees are commonly used to allow the efficient verification of transactions. Merkle root in a blockchain is present in the block header section of a block, which is the hash of all transactions in a block. This means that verifying only the Merkle root is required to verify all transactions present in the Merkle tree instead of verifying all transactions one by one.

- In addition to the block header, the block contains transactions that make up the block body. A **transaction** is a record of an event, for example, the event of transferring cash from a sender's account to a beneficiary's account. A block contains transactions, and its size varies depending on the type and design of the blockchain. For example, the Bitcoin block size is limited to one megabyte, which includes the block header of 80 bytes and transactions.

The following structure is a simple block diagram that depicts a generic block:

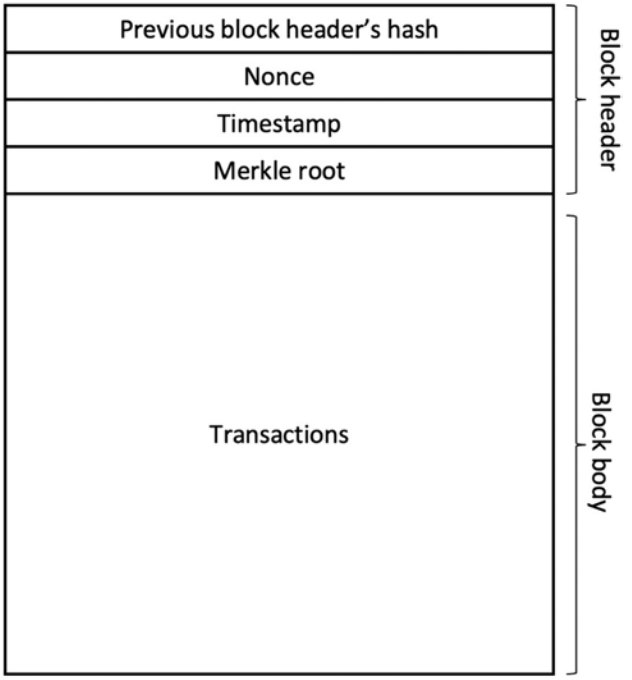

Figure 1.6: The generic structure of a block

Generally, there are just a few attributes that are essential to the functionality of a block: the block header, which is composed of the hash of the previous block's header, the timestamp, nonce, Merkle root, and the block body that contains the transactions. There are also other attributes in a block, but generally, the components introduced in this section are usually available in a block:

- **Peer-to-peer network**: As the name implies, a *peer-to-peer network* is a network topology wherein all peers can communicate with each other directly and send and receive messages.
- **The scripting or programming language**: *Scripts* or *programs* perform various operations on a transaction to facilitate various functions. For example, in Bitcoin, transaction scripts are predefined in a language called **Script**, which consists of sets of commands that allow nodes to transfer bitcoins from one address to another. Script is a limited language in the sense that it only allows essential operations that are necessary for executing transactions, but it does not allow arbitrary program development.

> Think of the scripting language as a calculator that only supports standard pre-programmed arithmetic operations. As such, the Bitcoin Script language cannot be called "Turing complete." In simple words, a Turing complete language means that it can perform any computation. It is named after Alan Turing, who developed the idea of a Turing machine, which can run any algorithm, however complex. Turing-complete languages need loops and branching capabilities to perform complex computations. Therefore, Bitcoin's scripting language is not Turing complete, whereas Ethereum's Solidity language is.
>
> To facilitate arbitrary program development on a blockchain, a Turing-complete programming language is needed, and it is now a very desirable feature to have for blockchains. Think of this as a computer that allows the development of any program using programming languages.

- **Virtual machine**: This is an extension of the transaction script introduced previously. A *virtual machine* allows Turing-complete code to be run on a blockchain (as smart contracts), whereas a transaction script is limited in its operation. However, virtual machines are not available on all blockchains. Various blockchains use virtual machines to run programs such as EVM and **Chain Virtual Machine** (**CVM**). EVM is used in the Ethereum blockchain, while CVM is a virtual machine developed for and used in an enterprise-grade blockchain called "Chain Core."
- **State machine**: A blockchain can be viewed as a state transition mechanism whereby a state is modified from its initial form to the next one by nodes on the blockchain network as a result of transaction execution.
- **Smart contracts**: These programs run on top of the blockchain and encapsulate the business logic to be executed when certain conditions are met. These programs are enforceable and automatically executable. The *smart contract* feature is not available on all blockchain platforms, but it is now becoming a very desirable feature due to the flexibility and power that it provides to blockchain applications. Smart contracts have many use cases, including but not limited to identity management, capital markets, trade finance, record management, insurance, and e-governance. Smart contracts will be discussed in more detail in *Chapter 8*, *Smart Contracts*.
- **Node**: A *node* in a blockchain network performs various functions depending on the role that it takes on. A node can propose and validate transactions and perform mining to facilitate consensus and secure the blockchain. This goal is achieved by following a **consensus protocol** (most commonly PoW). Nodes can also perform other functions, such as simple payment verification (lightweight nodes), validation, and many other functions depending on the type of blockchain used and the role assigned to the node. Nodes also perform a transaction signing function. Transactions are first created by nodes and then also digitally signed by nodes using private keys as proof that they are the legitimate owner of the asset that they wish to transfer to someone else on the blockchain network. This asset is usually a token or virtual currency, such as Bitcoin, but it can also be any real-world asset represented on the blockchain by using tokens. There are also now standards related to tokens; for example, on Ethereum, there are ERC20, ERC721, ERC777, and a few others that define the interfaces and semantics of tokenization.

A high-level diagram of blockchain architecture highlighting the key elements mentioned previously is shown as follows:

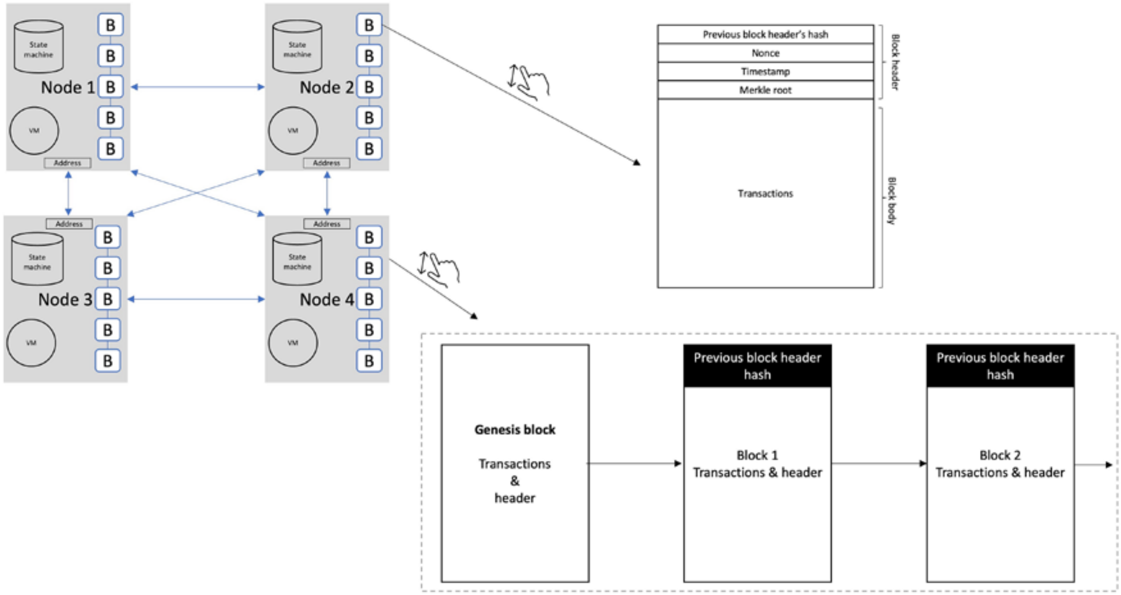

Figure 1.7: Generic structure of a blockchain network

The preceding diagram shows a four-node blockchain network (at the top), each maintaining a chain of blocks, virtual machine, state machine, and address. The blockchain is then further magnified (middle) to show the structure of the chain of blocks, which is again magnified (bottom) to show the structure of a transaction. Note that this is a generic structure of a blockchain; we will see specific blockchains structures in detail in the context of Ethereum and Bitcoin blockchains later in this book.

Blockchain functionality

We have now defined and described blockchain. Now, let's see how a blockchain works. Nodes are either *miners* that create new blocks and mint cryptocurrency (coins) or *block signers* that validate and digitally sign the transactions. A critical decision that every blockchain network must make is to figure out which node will append the next block to the blockchain. This decision is made using a *consensus mechanism*.

Consensus is a process of achieving agreement between distrusting nodes on the final state of data. To achieve consensus, different algorithms are used. It is easy to reach an agreement in a centralized network (in client-server systems, for example), but when multiple nodes are participating in a distributed system and they need to agree on a single value, it becomes quite a challenge to achieve consensus. This process of attaining agreement on a common state or value among multiple nodes is known as **distributed consensus**. If faults are allowed, then we call such a mechanism fault tolerant distributed consensus, where despite the failure of some nodes, agreement is reached between them.

Now, we will look at how a blockchain validates transactions and creates and adds blocks to grow the blockchain, using a general scheme for creating blocks:

1. **Transaction is initiated**: A node starts a transaction by first creating it and then digitally signing it with its private key. A transaction can represent various actions in a blockchain. Most commonly, this is a data structure that represents the transfer of value between users on the blockchain network. The transaction data structure usually consists of some logic of transfer of value, relevant rules, source and destination addresses, and other validation information. Transactions are usually either a cryptocurrency transfer (transfer of value) or a smart contract invocation that can perform any desired operation. A transaction occurs between two or more parties. This will be covered in more detail in specific chapters on Bitcoin and Ethereum later in the book.
2. **Transaction is validated and broadcast**: A transaction is propagated (broadcast) usually by using data-dissemination protocols, such as the Gossip protocol, to other peers that validate the transaction based on preset validity criteria. Before a transaction is propagated, it is also verified to ensure that it is valid.
3. **Find new block**: When the transaction is received and validated by special participants called miners on the blockchain network, it is included in a block, and the process of mining starts. This process is also sometimes referred to as "finding a new block." Here, nodes called miners race to finalize the block they've created by a process known as mining.
4. **New block found**: Once a miner solves a mathematical puzzle (or fulfills the requirements of the consensus mechanism implemented in a blockchain), the block is considered "found" and finalized. At this point, the transaction is considered confirmed. Usually, in cryptocurrency blockchains such as Bitcoin, the miner who solves the mathematical puzzle is also rewarded with a certain number of coins as an incentive for their effort and the resources they spent in the mining process.
5. **Add new block to the blockchain**: The newly created block is validated, transactions or smart contracts within it are executed, and it is propagated to other peers. Peers also validate and execute the block. It now becomes part of the blockchain (ledger), and the next block links itself cryptographically back to this block. This link is called a hash pointer.

This process can be visualized in the diagram as follows:

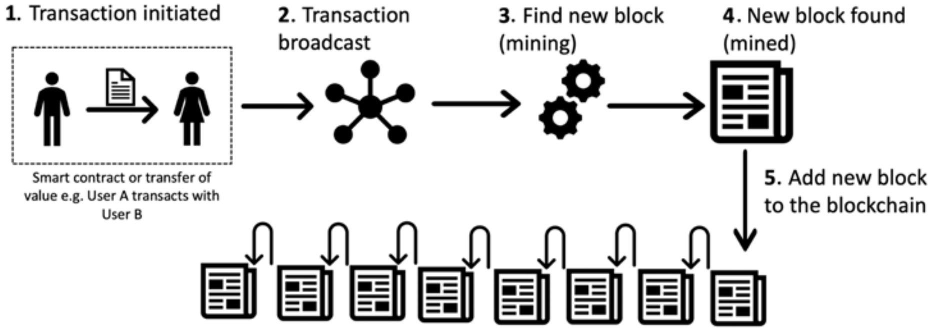
1. Transaction initiated
Smart contract or transfer of value e.g. User A transacts with User B
2. Transaction broadcast
3. Find new block (mining)
4. New block found (mined)
5. Add new block to the blockchain

Figure 1.8: How a block is generated

This completes the basic introduction to blockchain. In the next section, you will learn about the benefits and limitations of this technology.

Benefits and features of blockchain

Numerous advantages of blockchain technology have been discussed in many industries and proposed by thought leaders around the world who are participating in the blockchain space. The notable benefits of blockchain technology are as follows:

- **Simplification of current paradigms**: The current blockchain model in many industries, such as finance or health, is somewhat disorganized. In this model, multiple entities maintain their own databases and data sharing can become very difficult due to the disparate nature of the systems. However, as a blockchain can serve as a single shared ledger among many interested parties, this can result in simplifying the model by reducing the complexity of managing the separate systems maintained by each entity.
- **Faster dealings**: In the financial industry, especially in post-trade settlement functions, blockchain can play a vital role by enabling the quick settlement of trades. Blockchain does not require a lengthy process of verification, reconciliation, and clearance because a single version of agreed-upon data is already available on a shared ledger between financial organizations.
- **Cost-saving**: As no trusted third party or clearing house is required in the blockchain model, this can massively reduce overhead costs in the form of the fees, which are paid to such parties.
- **Smart property**: It is possible to link a digital or physical asset to the blockchain in such a secure and precise manner that it cannot be claimed by anyone else. You are in full control of your asset, and it cannot be double-spent or double-owned. Compare this with a digital music file, for example, which can be copied many times without any controls.

> While it is true that many **Digital Rights Management** (**DRM**) schemes are being used currently along with copyright laws, none of them are enforceable in the way a blockchain-based DRM can be. Blockchain can provide digital rights management functionality in such a way that it can be enforced fully: if you own an asset, no one else can claim it unless you decide to transfer it. This feature has far-reaching implications, especially in DRM and e-cash systems where double-spend detection is a crucial requirement.

- **Decentralization**: This is a core concept and benefit of blockchain. There is no need for a trusted third party or intermediary to validate transactions; instead, a consensus mechanism is used to agree on the validity of transactions.
- **Transparency and trust**: As blockchains are shared and everyone can see what is on the blockchain, this allows the system to be transparent. As a result, trust is established. This is more relevant in scenarios such as the disbursement of funds or benefits where personal discretion in relation to selecting beneficiaries needs to be restricted.

- **Immutability**: Once the data has been written to the blockchain, it is extremely difficult to change it back. It is not genuinely immutable, but because changing data is so challenging and nearly impossible, this is seen as a benefit to maintaining an immutable ledger of transactions and is especially useful in audit and compliance scenarios.
- **High availability**: As the system is based on thousands of nodes in a peer-to-peer network, and the data is replicated and updated on every node, the system becomes highly available. Even if some nodes leave the network or become inaccessible, the network continues to work, thus making it highly available. This redundancy results in high availability.
- **Highly secure**: All transactions on a blockchain are cryptographically secured and thus provide network integrity. Any transactions posted from the nodes on the blockchain are verified based on a predetermined set of rules. Only valid transactions are selected for inclusion in a block. The blockchain is based on proven cryptographic technology that ensures the integrity and availability of data. Generally, confidentiality is not provided due to the requirements of transparency. This limitation is the leading barrier to its adoption by financial institutions and other industries that require the privacy and confidentiality of transactions. As such, the privacy and confidentiality of transactions on the blockchain are being researched very actively, and advancements are already being made. It could be argued that, in many situations, confidentiality is not needed, and transparency is preferred. For example, with Bitcoin, confidentiality is not an absolute requirement; however, it is desirable in some scenarios. A more recent example is Zcash (`https://z.cash`), which uses zero-knowledge proofs to provide a platform for conducting anonymous transactions. Other security services, such as non-repudiation and authentication, are also provided by blockchain, as all actions are secured using private keys and digital signatures.
- **Platform for smart contracts**: Smart contracts are automated, autonomous programs that reside on the blockchain network and encapsulate the business logic and code needed to execute a required function when certain conditions are met. This is indeed a revolutionary feature of blockchain, as it provides flexibility, speed, security, and automation for real-world scenarios that can lead to a completely trustworthy system with significant cost reductions. Smart contracts can be programmed to perform any application-level actions that blockchain users need and according to their specific business requirements.

> Not all blockchains have a mechanism to execute *smart contracts*; however, this is a very desirable feature. However, note that some blockchains may not incorporate smart contract functionality on purpose, citing the reason that hardcoded executions are faster without the complexities of general-purpose smart contracts.

Limitations of blockchain technology

As with any technology, some challenges need to be addressed to make a system more robust, useful, and accessible. Blockchain technology is no exception. In fact, much effort is being made in both academia and industry to overcome the challenges posed by blockchain technology.

The most sensitive blockchain problems are as follows:

- **Scalability**: Currently, blockchain networks are not as scalable as, for example, current financial networks. This is a known area of concern and a very ripe area for research.
- **Regulation**: Due to its decentralized nature, regulation is almost impossible on blockchain. This is sometimes seen as a barrier toward adoption because, traditionally, due to the existence of regulatory authorities, consumers have a certain level of confidence that if something goes wrong they can hold someone accountable. However, in blockchain networks, no such regulatory authority and control exists, which is an inhibiting factor for many consumers.

> Note that there are attempts to regulate these networks, including the legalization of cryptocurrency exchanges in the US under the Bank secrecy act, so that these exchanges adhere to requirements such as **anti-money laundering** (AML), **Counter Financing of Terrorism** (CFT), and enforce strict record keeping and reporting to the authorities. Also, a promising use case enabled by blockchain is **central bank digital currency** (CBDC), which is a centralized and regulated form of digital currency issued by a central bank of a country.

- **Privacy**: Privacy is a concern on public blockchains such as Bitcoin where everyone can see every single transaction. This transparency is not desirable in many industries, such as the financial, law, or medical sectors. This is also a known concern and a lot of valuable research with some very impressive solutions has already been developed. A promising technology called zero-knowledge proofs is being utilized on blockchains to provide privacy. Further research continues to make these technologies better and more and more practical and mainstream.
- **Relatively immature technology**: As compared to traditional IT systems that have benefited from decades of research and evolution, blockchain is a comparatively new technology and requires research to achieve maturity. Even though core aspects of blockchains with the latest and most novel chains, such as Solana, Polkadot, and Avalanche, are very much mature; however, some features need to be improved further for example user experience, developer experience, security, and interoperability. From another angle, applications such as DeFi and Metaverses also need further maturity in terms of regulation, governance, and security.
- **Interoperability**: This problem is twofold. First is the interoperability of blockchains with enterprise and legacy systems, and second is the interoperability between different blockchains (cross-chain interoperability). Both these aspects are important to consider because blockchains cannot exist in silos; they must be able to communicate with one another, as well as enterprise networks and legacy systems, to enable enterprises to fully benefit from the technology. This is an active area of research, with many types of solutions becoming available in recent years, such as bridges, hub blockchains, and multi-chain hetereogeneous chains.

- **Adoption:** Often, blockchain is seen as a nascent technology. Even though this perspective has changed rapidly in the last few years, there is still a long way to go before the mass adoption of this technology. Some aspects, such as scalability, security, regulation, and customer confidence, need to be addressed before further adoption. Customer confidence can decrease due to security issues leading to loss of funds. Such problems may inhibit some users from joining the network, who would otherwise be happy to join if this security problem didn't exist. However, note that despite these problems, DeFi is becoming more and more popular, but it might discourage cautious customers.

All these issues and possible solutions will be discussed in detail later in this book.

You now know the basics of blockchain and its benefits and limitations. Now, let's look at the various types of blockchain that exist.

Types of blockchain

Based on the way that blockchain has evolved, it can be divided into multiple categories with distinct, though sometimes partially overlapping, attributes. At a broad level, **Digital Ledger Technology** (**DLT**) is an umbrella term that represents distributed ledger technology, comprising blockchains and distributed ledgers of different types:

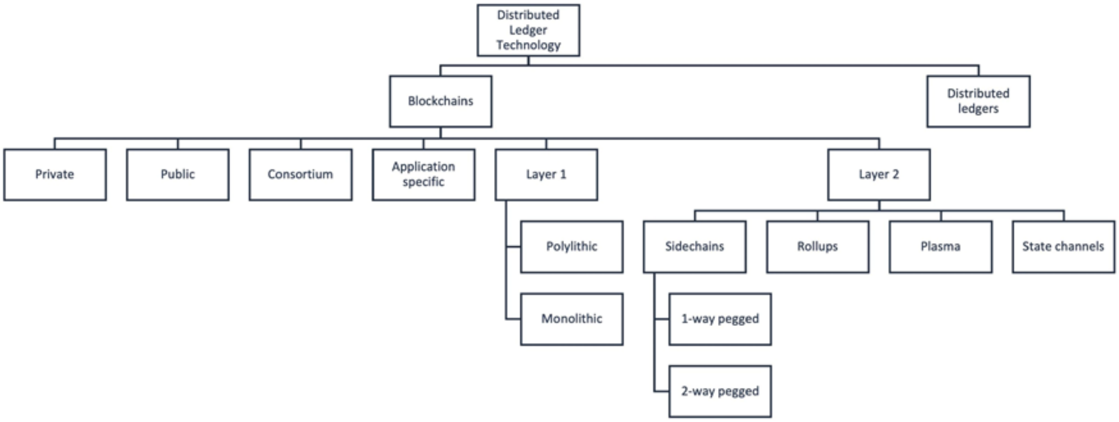

Figure 1.9: DLT types

In this section, we will examine the different types of blockchain from a technical and business perspective.

Distributed ledgers

A *distributed ledger* is a broad term describing shared databases; hence, all blockchains technically fall under the umbrella of shared databases or distributed ledgers. Although all blockchains are fundamentally distributed ledgers, not all distributed ledgers are necessarily blockchains.

A key difference between a distributed ledger and a blockchain is that a distributed ledger does not necessarily consist of blocks of transactions to keep the ledger growing. Rather, a blockchain is a special type of shared database that comprises blocks of transactions. An example of a distributed ledger that does not use blocks of transactions is R3's Corda (`https://www.corda.net`). Corda is a distributed ledger that is developed to record and manage agreements and is especially focused on the financial services industry. On the other hand, more widely known blockchains such as Bitcoin and Ethereum make use of blocks composed of transactions to update the replicated shared database. As the name suggests, a distributed ledger is distributed among its participants and is replicated across multiple nodes, sites, or organizations. This type of ledger can be either private or public.

Shared ledger

This is a generic term that is used to describe any application or database that is shared by the public or a consortium. Generally, all blockchains fall into the category of a shared ledger.

Public blockchains

As the name suggests, public blockchains are not owned by anyone. They are open to the public, and anyone can participate as a node. Users may or may not be rewarded for their participation. All users of these "permissionless" ledgers maintain a copy of the ledger on their local nodes and use a distributed consensus mechanism to decide the eventual state of the ledger. Bitcoin and Ethereum are both examples of public blockchains.

Private blockchains

As the name implies, private blockchains are just that—private. That is, they are open only to a consortium or group of individuals or organizations who have decided to share the ledger among themselves. There are various blockchains now available in this category, such as Hyperledger Fabric and Quorum. Optionally, both blockchains can also be in public mode if required, but their primary purpose is to provide a private blockchain. These blockchains are also called consortium blockchains, or enterprise blockchains.

Semi-private blockchains

With semi-private blockchains, part of the blockchain is private and part of it is public. Note that this is still just a concept today, and no real-world proofs of concept have yet been developed. With a semi-private blockchain, the private part is controlled by a group of individuals, while the public part is open for participation by anyone.

This hybrid model can be used in scenarios where the private part of the blockchain remains internal and shared among known participants, while the public part of the blockchain can still be used by anyone, optionally allowing mining to secure the blockchain. This way, the blockchain can be secured using PoW, thus providing consistency and validity for both the private and public segments. This type of blockchain can also be called a "semi-decentralized" model, where it is controlled by a single entity but still allows multiple users to join the network by following appropriate procedures.

Permissioned ledger

A *permissioned ledger* is a blockchain where participants of the network are already known and trusted. Permissioned ledgers do not need to use a distributed consensus mechanism; instead, an agreement protocol is used to maintain a shared version of the truth about the state of the records on the blockchain. In this case, for verification of transactions on the chain, all verifiers are already preselected by a central authority and, typically, there is no need for a mining mechanism.

There is no requirement for a permissioned blockchain to be private, as it can be a public blockchain but with regulated access control. For example, Bitcoin can become a permissioned ledger if an access control layer is introduced on top of it that verifies the identity of a user and then allows access to the blockchain.

Fully private and proprietary blockchains

There is no mainstream application of these types of blockchains, as they deviate from the core concept of decentralization in blockchain technology. Nonetheless, in specific private settings within an organization, there could be a need to share data and provide some level of guarantee of the authenticity of the data.

An example of this type of blockchain might be to allow collaboration and the sharing of data between various government departments. In that case, no complex consensus mechanism is required, apart from a simple SMR with known central validators. Even in private blockchains, tokens are not really required, but they can be used as a means of transferring value or representing some real-world assets.

Tokenized blockchains

These blockchains are standard blockchains that generate cryptocurrency as a result of a consensus process via mining or initial distribution. Bitcoin and Ethereum are prime examples of this type of blockchain.

Tokenless blockchains

These blockchains are designed in such a way that they do not have the basic unit for the transfer of value. However, they are still valuable in situations where there is no need to transfer value between nodes and only the sharing of data among various trusted parties is required. This is similar to fully private blockchains, the only difference being that the use of tokens is not required. This can also be thought of as a shared distributed ledger used for storing and sharing data between the participants. It does have its benefits when it comes to immutability, tamper-proofing, security, and consensus-driven updates, but is not used for a common blockchain application of value transfer or cryptocurrency. Most of the permissioned blockchains can be seen as an example of tokenless blockchains, for example, Hyperledger Fabric or Quorum. Tokens can be built on these chains as an application implemented using smart contracts, but intrinsically these blockchains do not have a token associated with them. In other words, we can say that there is no native (default) cryptocurrency in tokenless blockchains.

Layer 1 blockchains

Any base layer chain, responsible for consensus is a layer 1 blockchain. For example, Bitcoin and Ethereum. We can also think of two other types of blockchain architectures. One is **monolithic architecture**, which is just one base layer responsible for all operations, and another type is called **polylithic architecture**, which is composed of multiple chains.

Monolithic and polylithic blockchains

The original Bitcoin blockchain is a monolithic chain. Other examples include Ethereum and Solana. These chains are categorized as Layer 1 blockchains as they are base layer single-chain protocols where all functionalities including programmability (smart contracts), consensus protocol, security, and any related functionality are part of the same base blockchain. In other words, no component is off chain.

Polylithic chains include examples such as Polkadot, Avalanche, and Cosmos. In this type of architecture, multiple chains of the same or different types connect to a core chain and form a network of networks. Both types are considered layer 1 chains where a single base layer is the source of canonical truth. In polylithic architectures, there can be multiple chains, but they are horizontal to the core chain, and in some cases the core chain is not strictly needed and subnets can talk to each other directly. We can think of these architectures as multi-chain architectures. If chains connecting to the core chain are all the same type and built using the same rules, we call them homogeneous chains, and if they are of different types and follow different rules, we call them heterogeneous chains. Usually, multichain architectures aim to be heterogeneous architectures.

Layer 2 blockchains

Layer 2 blockchains have also recently emerged as a solution to the scalability and privacy problems on classical layer 1 blockchains, such as Ethereum and Bitcoin. Such solutions are called layer 2 solutions, which is a generic term used to describe solutions that use layer 1 as a base layer for consensus and settlement but execute transactions off chain at the so-called layer 2. These chains run on top of layer 1 chains. Many solutions exist in this space, such as sidechains, zero-knowledge rollups, optimistic rollups, plasma chains, and Lightning Network.

Sidechains

More precisely known as “pegged sidechains,” this is a concept whereby coins can be moved from one blockchain to another and then back again. Typical uses include the creation of new *altcoins* (alternative cryptocurrencies) whereby coins are burnt as a proof of an adequate stake. “Burnt” or “burning the coins” in this context means that the coins are sent to an address that is un spendable, and this process makes the “burnt” coins irrecoverable. This mechanism is used to bootstrap a new currency or introduce scarcity, which results in the increased value of the coin.

This mechanism is also called “Proof of Burn” and is used as an alternative method for distributed consensus to PoW and **Proof of Stake** (**PoS**). The example provided previously for burning coins applies to a **one-way pegged sidechain**. The second type is called a **two-way pegged sidechain**, which allows the movement of coins from the main chain to the sidechain and back to the main chain when required.

This process enables the building of smart contracts for the Bitcoin network. Rootstock is one of the leading examples of a sidechain, which enables smart contract development for Bitcoin using this paradigm. It works by allowing a two-way peg for the Bitcoin blockchain, and this results in much faster throughput. Another example is Deku, which is a side chain layer 2 solution for Tezos.

Summary

This chapter introduced blockchain technology at a high level. First, we discussed blockchain's progress toward becoming a mature technology, followed by some basic concepts about distributed systems, and then the history of blockchain was reviewed. Concepts such as e-cash were also discussed.

Various definitions of blockchain from different points of view were presented. Some applications of blockchain technology were also introduced. Next, different types of blockchain were explored. Finally, the benefits and limitations of this new technology were also examined. Some topics, such as blockchain scalability and adoptability issues, were intentionally introduced only lightly, as they will be discussed in depth in later chapters.

In the next chapter, we will introduce the concept of decentralization, which is central to the idea behind blockchains and their vast number of applications.

Join us on Discord!

To join the Discord community for this book – where you can share feedback, ask questions to the author, and learn about new releases – follow the QR code below:

`https://packt.link/ips2H`